



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

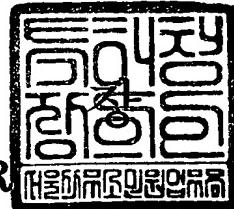
출원번호 : 10-2003-0009601  
Application Number

출원년월일 : 2003년 02월 15일  
Date of Application FEB 15, 2003

출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 08 월 22 일



특허청

COMMISSIONER

## 【서지사항】

【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【제출일자】	2003.02.15		
【발명의 명칭】	효율적인 패킷 메모리 관리 장치를 가지는 패킷 포워딩 시스템 및 그의 동작방법		
【발명의 영문명칭】	Packet forwarding system having efficient recevie data processor and a method processing thereof		
【출원인】			
【명칭】	삼성전자 주식회사		
【출원인코드】	1-1998-104271-3		
【대리인】			
【성명】	정홍식		
【대리인코드】	9-1998-000543-3		
【포괄위임등록번호】	2003-002208-1		
【발명자】			
【성명의 국문표기】	박우종		
【성명의 영문표기】	PARK,WOO JONG		
【주민등록번호】	710826-1074410		
【우편번호】	156-772		
【주소】	서울특별시 동작구 사당2동 극동아파트 112-1513		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 정홍식 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	8	면	8,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	10	항	429,000 원
【합계】	466,000 원		
【첨부서류】	1. 요약서·명세서(도면)_1통		

**【요약서】****【요약】**

효율적인 패킷 메모리 관리 장치를 가지는 패킷 포워딩 시스템 및 그의 동작방법이 개시된다. 패킷 포워딩 시스템은, 전송 단위로 데이터가 입력되는 입력부와, 데이터를 IP 패킷으로 조합하여 패킷 메모리에 기록하며 결정된 전송 순서대로 IP 패킷의 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하는 패킷 메모리 관리부와, 독출된 상기 헤더를 이용하여 패킷 분류 및 전송 목적지를 결정하고, 헤더에 연결된 상기 트레일러의 위치정보를 상기 패킷 메모리 관리부에 재전송하는 헤더처리부, 및 헤더처리부에서 전송된 헤더와 헤더에 연결된 트레일러의 위치정보에 기초하여 패킷 메모리 관리부에서 독출된 상기 트레일러을 전송 단위의 데이터로 분할하여 채널 경로로 출력하는 출력부를 갖는다.

**【대표도】**

도 4

**【색인어】**

패킷 메모리, 헤더, 트레일러, 패킷 포워딩

**【명세서】****【발명의 명칭】**

효율적인 패킷 메모리 관리 장치를 가지는 패킷 포워딩 시스템 및 그의 동작 방법{Packet forwarding system having efficient recevie data processor and a method processing thereof}

**【도면의 간단한 설명】**

도 1은 종래의 패킷 메모리 관리 시스템에 대한 블록도,  
도 2는 도 1의 전송 큐에 대한 개념도,  
도 3은 본 발명에 따른 패킷 포워딩 시스템에 대한 개략적인 블록도,  
도 4는 도 3의 패킷 메모리 관리부(400)에 대한 개략적인 블록도,  
도 5는 도 4의 엠프티 큐에 대한 구조 및 특성을 설명하기 위한 개념도,  
도 6은 도 4의 전송 헤더큐에 대한 구조 및 특성을 설명하기 위한 개념도,  
도 7a는 도 4의 패킷 메모리에 대한 구조도,  
도 7b는 도 4의 버퍼내 기록되는 디스크립터 및 태그 포맷을 나타낸 도,  
도 8은 본 발명에 따른 패킷 메모리의 관리방법을 설명하기 위한 개념도,  
그리고,  
도 9는 본 발명에 따른 패킷 메모리의 관리방법에 대한 흐름도이다.

\* 도면의 주요 부분에 대한 부호의 설명 \*

400 : 패킷 메모리 관리부    410 : 패킷조합부

430 : 컨트롤러    450 : 엠프티 큐

470 : 전송 헤더큐      490 : 패킷 메모리

491 : 트레일러큐      600 : 헤더처리부

800 : 출력부

### 【발명의 상세한 설명】

### 【발명의 목적】

### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <17> 본 발명은 전송된 패킷을 목적지에 고속으로 전달하기 위한 패킷 포워딩 시스템(Packet Forwarding system)에 관한 것으로, 보다 상세하게는 효율적인 패킷 메모리 관리 장치를 가지는 패킷 포워딩 시스템 및 그의 동작방법에 관한 것이다.
- <18> 도 1은 종래의 패킷 포워딩 시스템의 패킷 메모리 관리 장치에 대한 개략적인 블록도이다.
- <19> 패킷 메모리 관리 장치는 패킷조합부(110), 컨트롤러(130), 엠프티 큐(150), 전송 큐(170), 및 패킷 메모리(190)를 갖는다.
- <20> 패킷 조합부(110)는 데이터 전송 단위(예컨대, ATM 셀)로 입력되는 데이터를 조합하여 IP 패킷(Internet Protocol packet)을 생성하며, IP 패킷은 적어도 하나 이상의 일정한 길이를 갖는 서브 패킷(sub packet)으로 조합된다.
- <21> 컨트롤러(130)는 서브 패킷을 패킷 메모리(190)에 리드 및 라이트하기 위해 엠프티 큐(150) 및 전송 큐(170)를 제어하며, 조합된 IP 패킷을 정해진 전송 순서대로 전달되도록 전반적인 패킷 메모리(190)를 제어한다.

- <22> 엠프티 큐(Empty Queue)(150)는 패킷 메모리(190)의 빈 방 즉, 빈 버퍼의 위치 정보를 가지며, 엠프티 큐(150)는 헤드 엔트리(head entry)부터 트레일러 엔트리(tail entry)까지의 상호간에 연결정보를 갖는 스택(stack)형태로 동작한다.
- <23> 전송 큐(TX Queue)(170)는 IP 패킷의 전송 순서에 대응하여 IP 패킷의 헤더 및 트레일러의 위치정보가 기록되며 FIFO 형태로 동작한다.
- <24> 도 3은 도2의 종래의 전송 큐(170)에 대한 개념도이다. 전송 큐(170)는 최대 N개의 IP 패킷 헤더의 위치정보를 기록할 공간이 필요하고, 또한, IP 패킷의 최대 크기인 트레일러의 위치정보를 기록할 공간이 필요하다. 즉, 컨트롤러(130)는 전송 큐(170)에 기록된 IP 패킷의 헤더 및 트레일러 위치정보에 기초하여 패킷 메모리(190)에 기록된 IP 패킷의 헤더 및 트레일러를 각각 전송한다.
- <25> 일반적으로 엠프티 큐(150)와 전송 큐(170)의 저장매체는 랜덤 동작에 성능이 우수한 SRAM을 사용하며, 패킷 메모리(190)의 저장매체는 버스트 동작에 성능이 우수한 DRAM을 사용한다. 즉, IP 패킷이 패킷 메모리(190)에 기록되는 특성을 보면, IP 패킷의 헤더는 랜덤하게 기록되나, IP 패킷의 헤더에 따른 트레일러는 상호 연결정보를 갖고 버스트하게 기록된다.
- <26> 따라서, 저장매체의 특성 및 저장매체에 기록되는 데이터의 특성에 따라서 종래의 전송 큐(170)의 사용 효율이 저하되는 문제점을 갖는다.
- <27> 또한, 비용면에서도 큰 용량의 전송 큐, SRAM을 사용함에 따라서 역시, 비효율적이다.

**【발명이 이루고자 하는 기술적 과제】**

<28> 상기와 같은 문제점을 해결하기 위한 본 발명에 따른 목적은, 패킷 메모리에 숨겨진 IP 패킷의 트레일러 큐를 제안함으로써 효율적인 패킷 메모리 관리가 가능한 패킷 메모리 관리 장치를 가지는 패킷 포워딩 시스템 및 그의 동작방법을 제공하는 것이다.

**【발명의 구성 및 작용】**

<29> 상기 목적을 달성하기 위한 본 발명에 따른 효율적인 패킷 메모리 관리 장치를 가지는 패킷 포워딩 시스템은, 전송 단위로 데이터가 입력되는 입력부와, 상기 데이터를 IP 패킷으로 조합하여 패킷 메모리에 기록하며 결정된 전송 순서대로 상기 IP 패킷의 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하는 패킷 메모리 관리부와, 독출된 상기 헤더를 이용하여 패킷 분류 및 전송 목적지를 결정하고, 상기 헤더에 연결된 상기 트레일러의 위치정보를 상기 패킷 메모리 관리부에 재전송하는 헤더처리부, 및 상기 헤더처리부에서 전송된 상기 헤더와 상기 헤더에 연결된 상기 트레일러의 위치정보에 기초하여 상기 패킷 메모리 관리부에서 독출된 상기 트레일러를 상기 전송 단위의 데이터로 분할하여 채널 경로로 출력하는 출력부를 갖는다.

<30> 상기 패킷 메모리 관리부는, 입력되는 데이터를 이용하여 상기 IP 패킷으로 조합하는 패킷조합부와, 상기 IP 패킷을 기록하는 다수의 버퍼를 가지며 상기 버퍼 내에는 상기 버퍼의 속성정보 및 상기 헤더와 연결된 상기 트레일러의 위치정보가 기록되는 패킷 메모리와, 상기 IP 패킷의 전송순서에 대응하여 상기 헤더의 위치정보가 기록되는 전송 헤더큐, 및 상기 전송 헤더큐에서 의해 결정된 전송

순서에 대응하여 상기 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하여 상기 헤더처리부에 전송하는 컨트롤러를 갖는다.

<31> 상기 컨트롤러는, 상기 헤더처리부로부터 상기 헤더와 연결된 상기 트레일러의 위치정보가 재전송되는 경우 상기 트레일러의 위치정보에 대응하는 버퍼에서 상기 헤더와 연결된 상기 트레일러를 독출하여 상기 출력부에 전송한다.

<32> 또한, 상기 컨트롤러는, 상기 트레일러의 위치정보에 대응하는 버퍼의 속성 정보를 이용하여 상기 트레일러와 연결되는 다른 트레일러의 유무를 확인하여 상기 다른 트레일러가 존재하는 경우 상기 다른 트레일러를 독출하여 상기 출력부에 전송한다.

<33> 상기 버퍼의 속성정보는, 상기 버퍼와 전후에 연결된 버퍼의 위치정보, 및 상기 버퍼에 기록된 상기 트레일러 다음에 연결되는 상기 다른 트레일러가 존재 여부에 대한 정보를 갖는다.

<34> 한편, 본 발명에 따른 패킷 포워딩 방법은, 전송 단위로 데이터가 입력되는 단계; 상기 데이터를 IP 패킷으로 조합하여 패킷 메모리에 기록하며, 결정된 전송 순서대로 상기 IP 패킷의 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하여 전송하는 패킷 메모리 관리단계; 독출된 상기 헤더를 이용하여 패킷 분류 및 전송 목적지를 결정하고, 상기 헤더에 연결된 상기 트레일러의 위치정보를 상기 패킷 메모리 관리단계에 재전송하는 헤더처리단계; 및 상기 헤더처리단계에서 전송된 상기 헤더와 상기 헤더에 연결된 상기 트레일러의 위치정보에 기초하여 상기 패킷 메모리 관리단계에서 독출된 상기 트레

일러을 상기 전송 단위의 데이터로 분할하여 채널 경로로 출력하는 출력단계;를 가지는 것을 특징으로 한다.

<35> 상기 패킷 메모리 관리단계는, 입력되는 데이터를 이용하여 상기 IP 패킷으로 조합하는 단계; 상기 IP 패킷을 다수의 버퍼에 기록하며 상기 버퍼 내에는 상기 버퍼의 속성정보 및 상기 헤더와 연결된 상기 트레일러의 위치정보를 기록하는 단계; 결정된 전송순서에 대응하여 상기 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하여 상기 헤더처리단계로 전송하는 단계;를 갖는다.

<36> 상기 패킷 메모리 관리단계는, 상기 헤더처리단계로부터 상기 헤더와 연결된 상기 트레일러의 위치정보가 재전송되는 경우, 상기 트레일러의 위치정보에 대응하는 버퍼에서 상기 헤더와 연결된 상기 트레일러를 독출하여 상기 출력단계에 전송하는 단계;를 갖는다.

<37> 또한, 상기 패킷 메모리 관리단계는, 상기 트레일러의 위치정보에 대응하는 버퍼의 속성정보를 이용하여 상기 트레일러와 연결되는 다른 트레일러의 유무를 확인하여 상기 다른 트레일러가 존재하는 경우 상기 다른 트레일러를 독출하여 상기 출력단계에 전송하는 단계;를 갖는다.

<38> 따라서, 헤더 큐 및 트레일러 큐의 동작 특성에 맞는 저장매체를 사용함으로써 패킷 메모리 관리를 보다 효율적으로 수행할 수 있다. 또한, 고속의 패킷 처리가 필요한 패킷 포워딩 시스템에서 헤더처리에 걸리는 시간적 문제로 인한 성능저하를 해결할 수 있다.

- <39> 이하에서는 도면을 참조하여 본 발명을 보다 상세하게 설명한다.
- <40> 도 3은 본 발명에 따른 패킷 포워딩 시스템(Packet forwarding system)에 대한 개략적인 블록도이다. 패킷 포워딩 시스템은 입력부(200), 패킷 메모리 관리부(400), 헤더처리부(600), 및 출력부(800)로 구성된다.
- <41> 입력부(200)는 외부 디바이스와 패킷 포워딩 시스템이 연결되는 인터페이스이며, 소정의 데이터 전송 단위로 데이터가 입력된다. 여기서, 소정의 데이터 전송 단위는, 예컨데, ATM 셀(cell)단위이다.
- <42> 패킷 메모리 관리부(400)는 입력되는 전송 데이터 단위의 데이터, 예컨데, ATM 셀의 헤더를 제거하고, 헤더가 제거된 ATM 셀을 조합하여 IP 패킷을 생성한다. 생성된 IP 패킷을 관리하여 전송 순서를 정하고 정해진 전송 순서대로 IP 패킷을 출력한다. 본 발명에 따른 패킷 메모리 관리부(400)는 상세하게 후술된다.
- <43> 헤더처리부(600)는 패킷 메모리 관리부(400)로부터 제공되는 IP 패킷의 헤더를 이용하여 패킷 분류 및 전송할 목적지를 결정한다. IP 패킷의 헤더에는 목적지 어드레스, 발신지 어드레스 또는 기타 데이터에 따라 다양한 형태의 서비스가 제공된다. IP 패킷의 헤더에 포함된 목적지 어드레스를 이용하여 전송할 목적지를 결정한다.
- <44> 출력부(800)는 헤더처리부(600) 및 패킷 메모리 관리부(800)에서 각각 제공되는 IP 패킷의 헤더 및 트레일러를 소정의 데이터 전송 단위, 예컨데, ATM 셀 단위로 분할한다. 분할된 각각의 ATM 셀에 ATM 헤더를 생성하여 채널 경로로 출력한다.

- <45> 도 4는 본 발명에 따른 패킷 메모리 관리부(400)에 대한 상세한  
블록도이며, 도 5 내지 도 7b를 참조하여 상세하게 설명한다.
- <46> 패킷 메모리 관리부(400)는 패킷 조합부(410), 컨트롤러(430), 엠프티 큐  
(450), 전송 헤더큐(470), 및 패킷 메모리(490)를 갖는다.
- <47> 패킷 조합부(410)는 데이터 전송 단위(예컨대, ATM 셀)로 입력되는 데이터  
를 조합하여 IP 패킷을 생성하며, IP 패킷은 적어도 하나 이상의 일정한 길이를  
갖는 서브 패킷(sub packet)을 갖는다.
- <48> 컨트롤러(430)는 서브 패킷을 패킷 메모리(490)에 리드 및 라이트하기 위해  
엠프티 큐(450) 및 전송 헤더큐(470)를 제어하며, 전송 헤더큐(470)에 의해 정해  
진 IP 패킷의 헤더를 전송 순서대로 헤더처리부(600)로 전송한다. 또한, 헤더와  
연결되는 트레일러의 위치정보를 가지는 태그를 생성하여 헤더처리부(600)에 헤  
더를 전송할 때, 해당 헤더의 태그를 함께 전송한다.
- <49> 엠프티 큐(450)는 패킷 메모리(490)의 빈 방 즉, 빈 버퍼의 위치 정보를 갖  
는다. 엠프티 큐(450)의 구조 및 특성은, 도 5에 도시된 바와 같이, 엠프티 큐  
(450)의 헤드 엔트리 Empty\_Link\_Head(0h)부터 테일 엔트리  
Empty\_Link\_Tail(N-1h)까지의 상호 연결정보를 갖는 스택(stack)형태로  
동작한다.
- <50> 전송 헤더큐(470)는 IP 패킷의 전송 순서에 대응하는 IP 패킷의 헤더의 위  
치 정보가 기록된다. 전송 헤더큐(470)의 구조 및 특성은, 도 6에 도시된 바와

같이, 전송해야 할 IP 패킷의 헤더 정보가 FIFO 형태로 동작한다. 한편, 트레일러 큐(491)는 패킷 메모리(490)에 포함된다.

<51> 패킷 메모리(490)는 다수의 버퍼를 가지며, 엠프티 큐(450)에서 제공되는 빈 버퍼의 위치정보에 기초하여 IP 패킷은 서브 패킷 단위로 기록된다. 한편, 트레일러 큐(491)를 갖는다. 패킷 메모리(490)에 포함된 트레일러 큐(491)는 도 7a 및 도 7b를 참고하여 설명한다.

<52> 도 7a 및 도 7b는 패킷 메모리(490)에 대한 일 예를 도시한 도이다. 도 7a 는 버퍼(B) 크기가 512byte인 N개(N= 128K)의 버퍼(B)를 갖는 패킷 메모리(490)를 도시한 것이며, 도 7b는 각각의 버퍼(B)에 포함된 디스크립터(D:descriptor) 및 태그(T:tag)에 대한 포맷을 도시한 것이다. 도 7a 및 도 7b에 도시된 패킷 메모리(490)에 포함된 트레일러 큐(491)에 대한 설명을 위한 것이며, 구조 및 설계는 다양하게 될 수 있다.

<53> 도 7b에 도시된 바와 같이, 태그(T) 내에는 트레일러의 위치 정보(Trailer buffer pointer)가 포함되며, 디스크립터(D) 내에는 해당 버퍼 이후에 트레일러가 존재하는지에 대한 정보 N(Next Trailer exist)를 갖는다. 즉, 해당 버퍼 이후에 트레일러가 존재하는 경우, 다음의 버퍼 위치 정보(next buffer pointer)는 이후의 트레일러가 존재하는 버퍼의 위치 정보가 된다. 이와 같은 방식에 의해 패킷 메모리(490)내의 트레일러 큐(491)가 동작된다.

<54> 따라서, 패킷 메모리(490)에 사용되는 버스트한 동작 특성을 갖는 DRAM을 사용하여 버스트한 동작 특성을 갖는 트레일러 큐의 동작시킴으로써 보다 효율적

으로 패킷 메모리를 관리할 수 있게 된다. 또한, 전송 헤더큐(470)에 사용되는 비용이 비싼 SRAM의 용량을 줄임으로써 경제적인 면에서도 보다 효율적이다.

<55> 더불어, 종래의 전송 큐 방식을 갖는 패킷 포워딩 시스템은 헤더처리부(600)에서 IP 패킷의 헤더 뿐만아니라 사용되지 않는 트레일러까지 전송됨에 따른 헤더처리에 걸리는 불필요한 시간소비에 따른 성능저하의 문제를 해결할 수 있게 된다. 즉, 본 발명에 따른 전송 헤더큐(470)는 IP 패킷의 헤더 만을 헤더처리부(600)에 전송하며, 함께 전송된 태그에 의해 헤더와 연결된 트레일러를 패킷 메모리(490)에서 독출하여 직접 출력부(800)로 전송함으로써 헤더처리에 걸리는 불필요한 시간소비를 막을 수 있게 된다.

<56> 도 8 및 도 9를 참조하여 본 발명에 따른 패킷 포워딩 시스템에서 효율적으로 패킷 메모리를 관리하는 방법을 상세하게 설명한다.

<57> 패킷 조합부(410)에서 조합된 IP 패킷(A)은 서브 패킷 A1, A2, A3로 이루어 진다(S811). 컨트롤러(430)는 엠프트 큐(450)에서 제공하는 패킷 메모리(490)의 버퍼 중 빈 버퍼의 위치 정보에 기초하여 서브 패킷 A1, A2, A3을 할당된 버퍼(B0, B2, B6)에 각각 기록한다. 이때, 각각의 버퍼(B0, B2, B6)내의 디스크립터 및 태그에 서브 패킷 A1, A2, A3의 상호 연결정보를 기록한다(S813).

<58> 즉, IP 패킷의 헤더인 서브 패킷 A1의 태그에는 연결된 트레일러인 서브 패킷 A2이 기록된 버퍼 B2의 위치 정보를 기록한다. 또한, 트레일러 A2가 기록된 버퍼 B2의 디스크립터에는 이후에 연결된 트레일러의 존재여부에 대한 정보(N)를 기록하고, 버퍼 B2에 연결된 버퍼 B6의 위치 정보를 기록한다. 이와 같이, 패킷 메모리(490)에 기록되는 서브 패킷 A1, A2, A3는 상호 연결 정보를 갖는다.

- <59> 패킷 메모리(490)에 기록된 서브 패킷 A1, A2, A3 중 IP 패킷의 헤더인 A1의 위치 정보는 전송 헤더큐(470)에 기록된다.
- <60> 이후, 이상과 같은 과정을 반복하여 전송 헤더큐(470)에 전송 순서가 결정된 헤더의 위치 정보가 모두 기록되면 컨트롤러(430)는 전송 헤더큐(470)의 헤드 엔트리의 위치 정보를 이용하여 해당하는 헤더를 독출하여 헤더처리부(600)에 전송한다(S815). 예컨대, 전송 헤더큐(470)의 헤드 엔트리가 헤더 A1이 기록된 버퍼 B0의 위치 정보인 경우, 컨트롤러(430)는 헤더 A1을 헤더처리부(600)에 전송하고, 또한, 버퍼 B0의 태그를 헤더 A1과 함께 헤더처리부(600)에 전송한다. 헤더 A1이 헤더처리부(600)에 전송됨과 동시에 컨트롤러(430)는 헤더 A1이 기록되었던 버퍼 B0의 위치 정보를 앰프터 큐(450)에 업데이트한다.
- <61> 이 후, 헤더처리부(600)에서는 IP 패킷(A)의 헤더 A1에 포함된 다양한 정보를 이용하여 패킷 분류 및 전송할 목적지를 결정한다. 헤더처리부(600)의 동작이 종료되면, 헤더처리부(600)는 태그에 포함된 헤더 A1과 연결된 트레일러 A2가 기록된 버퍼 B2의 위치 정보(Trailer buffer pointer)를 패킷 메모리 관리부(400)의 컨트롤러(430)에 재전송함으로써 헤더 A1에 연결된 트레일러 A2를 요청하게 된다(S817). 물론, 헤더에 연결된 트레일러가 없을 경우에는 트레일러 요청 이하의 동작은 생략된다.
- <62> 컨트롤러(430)는 재전송된 버퍼 B2의 위치 정보를 이용하여 트레일러 A2를 독출하고, 버퍼 B2의 디스크립터에 포함된 N(Next Trailer exist) 및 next buffer pointer를 이용하여 버퍼 B6의 위치 정보를 얻을 수 있다. 이에 의해 버

며 B6에 기록된 트레일러 A3를 독출한다(S819). 이때, 컨트롤러(430)는 베퍼 B2 및 B6의 위치 정보는 엠프티 큐(450)에 업데이트한다.

<63> 이렇게 패킷 메모리(490)에 포함된 트레일러 큐에 의해 독출된 트레일러 A2, A3는 출력부(800)에 전송되며, 헤더 A1은 헤더처리부(600)에서 처리된 후 출력부(800)에 전송된다.

<64> 출력부(800)는 제공된 IP 패킷을 데이터 전송 단위로 분할하여 채널 경로로 출력한다.

<65> 따라서, 헤더 큐 및 트레일러 큐의 동작 특성에 맞는 저장매체를 사용함으로써 패킷 메모리 관리를 보다 효율적으로 수행할 수 있다.

<66> 또한, 고속의 패킷 처리가 필요한 패킷 포워딩 시스템에서 헤더처리에 걸리는 시간적 문제로 인한 성능저하를 해결할 수 있다.

### 【발명의 효과】

<67> 본 발명에 따르면, 첫째, 패킷 메모리의 관리를 보다 효율적으로 할 수 있고, 둘째, 패킷 메모리의 구현에 있어 비용을 절감할 수 있다.

<68> 세째로, 고속의 패킷 처리가 필요한 패킷 포워딩 시스템에서 헤더처리에 걸리는 시간적 문제로 인한 성능저하를 해결할 수 있다. 이에 따라서, 전체 IP 패킷 중 헤더 만을 헤더처리부에 보내어 헤더를 처리함으로써 고속으로 패킷을 포워딩할 수 있다. 이에 의해 라우터, 스위치, 게이트웨이 등에서 향상된 성능을 얻을 수 있다.

<69> 이상에서는 본 발명의 바람직한 실시예에 대해 도시하고 설명하였으나, 본 발명은 상술한 특정의 바람직한 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능한 것은 물론이고, 그와 같은 변경은 청구범위 기재의 범위 내에 있게 된다.

**【특허청구범위】****【청구항 1】**

전송 단위로 데이터가 입력되는 입력부;

상기 데이터를 IP 패킷(Internet Protocol packet)으로 조합하여 패킷 메모리에 기록하며, 결정된 전송 순서대로 상기 IP 패킷의 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하는 패킷 메모리 관리부;

독출된 상기 헤더를 이용하여 패킷 분류 및 전송 목적지를 결정하고, 상기 헤더에 연결된 상기 트레일러의 위치정보를 상기 패킷 메모리 관리부에 재전송하는 헤더처리부; 및

상기 헤더처리부에서 전송된 상기 헤더와, 상기 헤더에 연결된 상기 트레일러의 위치정보에 기초하여 상기 패킷 메모리 관리부에서 독출된 상기 트레일러를 상기 전송 단위의 데이터로 분할하여 채널 경로로 출력하는 출력부;를 포함하는 것을 특징으로 패킷 포워딩 시스템.

**【청구항 2】**

제 1항에 있어서,

상기 패킷 메모리 관리부는,

입력되는 데이터를 이용하여 상기 IP 패킷으로 조합하는 패킷조합부;

상기 IP 패킷을 기록하는 다수의 버퍼를 가지며, 상기 버퍼 내에는 상기 버퍼의 속성정보 및 상기 헤더와 연결된 상기 트레일러의 위치정보가 기록되는 패킷 메모리;

상기 IP 패킷의 전송순서에 대응하여 상기 헤더의 위치정보가 기록되는 전송 헤더큐; 및

상기 전송 헤더큐에서 의해 결정된 전송순서에 대응하여 상기 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하여 상기 헤더처리부에 전송하는 컨트롤러;를 포함하는 것을 특징으로 하는 패킷 포워딩 시스템.

#### 【청구항 3】

제 2항에 있어서,

상기 컨트롤러는,

상기 헤더처리부로부터 상기 헤더와 연결된 상기 트레일러의 위치정보가 재전송되는 경우, 상기 트레일러의 위치정보에 대응하는 버퍼에서 상기 헤더와 연결된 상기 트레일러를 독출하여 상기 출력부에 전송하는 것을 특징으로 하는 패킷 포워딩 시스템.

#### 【청구항 4】

제 2항에 있어서,

상기 컨트롤러는,

상기 트레일러의 위치정보에 대응하는 버퍼의 속성정보를 이용하여 상기 트레일러와 연결되는 다른 트레일러의 유무를 확인하여 상기 다른 트레일러가 존재하는 경우 상기 다른 트레일러를 독출하여 상기 출력부에 전송하는 것을 특징으로 하는 패킷 포워딩 시스템.

#### 【청구항 5】

제 2항에 있어서,

상기 버퍼의 속성정보는,

상기 버퍼와 전후에 연결된 버퍼의 위치정보, 및 상기 버퍼에 기록된 상기 트레일러 다음에 연결되는 상기 다른 트레일러가 존재여부에 대한 정보를 포함하는 것을 특징으로 하는 패킷 포워딩 시스템.

#### 【청구항 6】

전송 단위로 데이터가 입력되는 단계;

상기 데이터를 IP 패킷으로 조합하여 패킷 메모리에 기록하며, 결정된 전송 순서대로 상기 IP 패킷의 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하여 전송하는 패킷 메모리 관리단계;

독출된 상기 헤더를 이용하여 패킷 분류 및 전송 목적지를 결정하고, 상기 헤더에 연결된 상기 트레일러의 위치정보를 상기 패킷 메모리 관리단계에 재전송하는 헤더처리단계; 및

상기 헤더처리단계에서 전송된 상기 헤더와, 상기 헤더에 연결된 상기 트레일러의 위치정보에 기초하여 상기 패킷 메모리 관리단계에서 독출된 상기 트레일러

러을 상기 전송 단위의 데이터로 분할하여 채널 경로로 출력하는 출력단계;를 포함하는 것을 특징으로 패킷 포워딩 방법.

### 【청구항 7】

제 6항에 있어서,

상기 패킷 메모리 관리단계는,

입력되는 데이터를 이용하여 상기 IP 패킷으로 조합하는 단계;

상기 IP 패킷을 다수의 버퍼에 기록하며, 상기 버퍼 내에는 상기 버퍼의 속성정보 및 상기 헤더와 연결된 상기 트레일러의 위치정보를 기록하는 단계; 및 결정된 전송순서에 대응하여 상기 헤더 및 상기 헤더와 연결된 트레일러의 위치정보를 상기 패킷 메모리에서 독출하여 상기 헤더처리단계로 전송하는 단계; 를 포함하는 것을 특징으로 하는 패킷 포워딩 방법.

### 【청구항 8】

제 7항에 있어서,

상기 패킷 메모리 관리단계는,

상기 헤더처리단계로부터 상기 헤더와 연결된 상기 트레일러의 위치정보가 재전송되는 경우, 상기 트레일러의 위치정보에 대응하는 버퍼에서 상기 헤더와 연결된 상기 트레일러를 독출하여 상기 출력단계에 전송하는 단계;를 더 포함하는 것을 특징으로 하는 패킷 포워딩 방법.

### 【청구항 9】

제 7항에 있어서,

상기 패킷 메모리 관리단계는,

상기 트레일러의 위치정보에 대응하는 버퍼의 속성정보를 이용하여 상기 트레일러와 연결되는 다른 트레일러의 유무를 확인하여 상기 다른 트레일러가 존재하는 경우 상기 다른 트레일러를 독출하여 상기 출력단계에 전송하는 단계;를 더 포함하는 것을 특징으로 하는 패킷 포워딩 방법.

#### 【청구항 10】

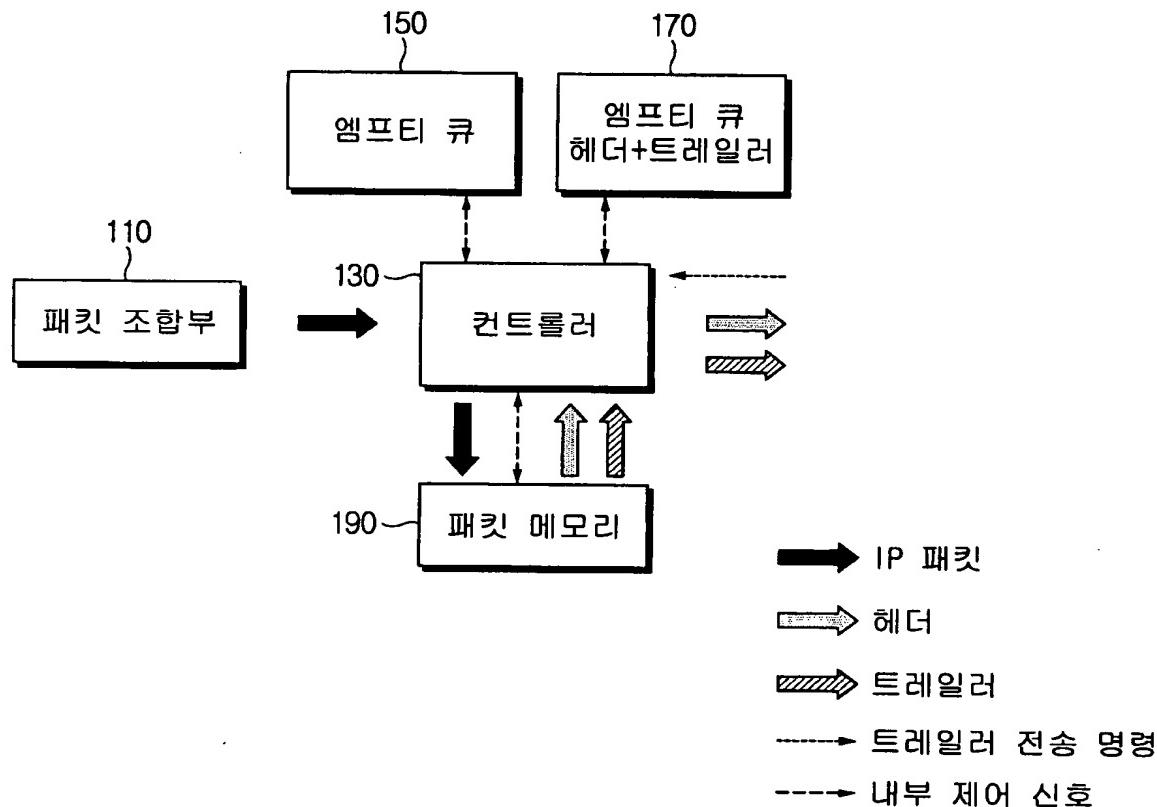
제 7항에 있어서,

상기 버퍼의 속성정보는,

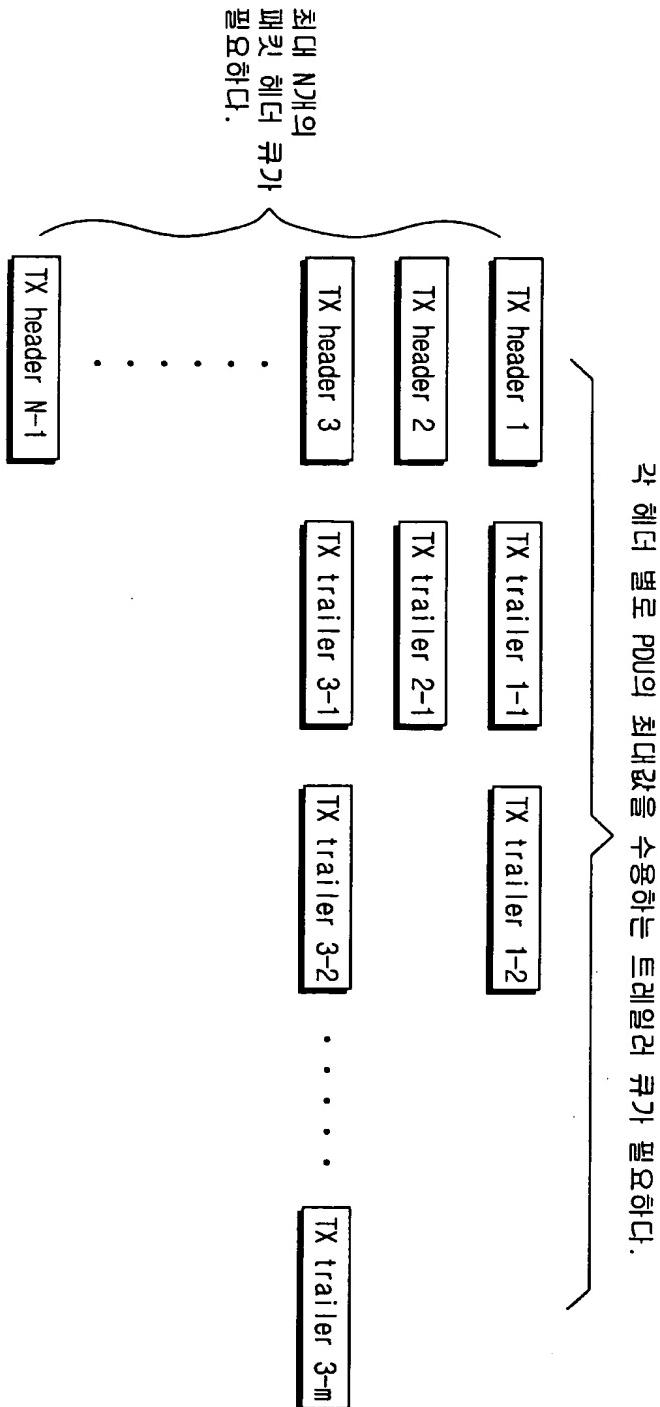
상기 버퍼와 전후에 연결된 버퍼의 위치정보, 및 상기 버퍼에 기록된 상기 트레일러 다음에 연결되는 상기 다른 트레일러가 존재여부에 대한 정보를 포함하는 것을 특징으로 하는 패킷 포워딩 방법.

## 【도면】

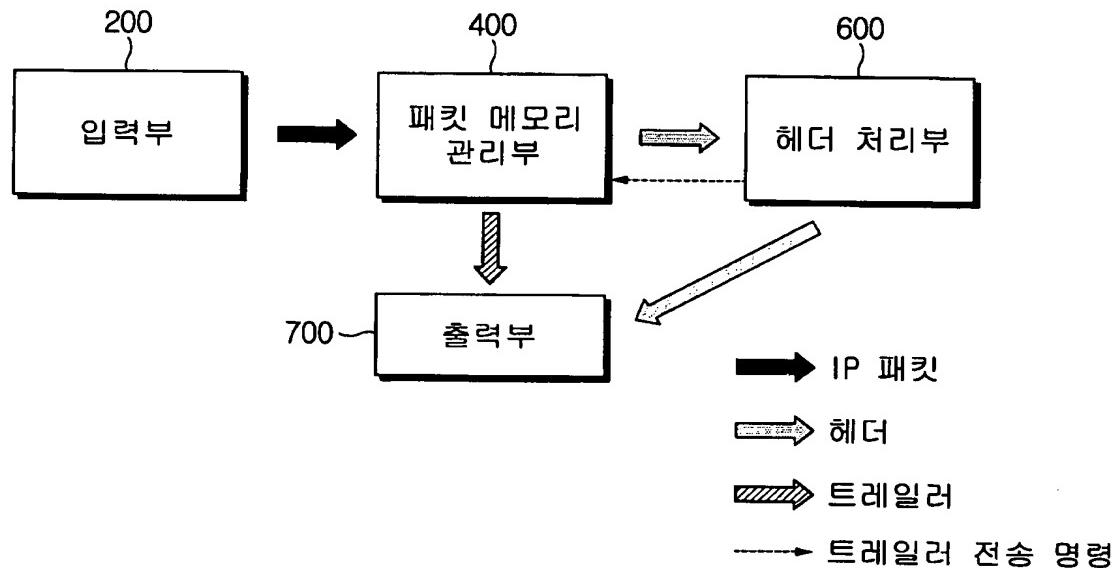
【도 1】



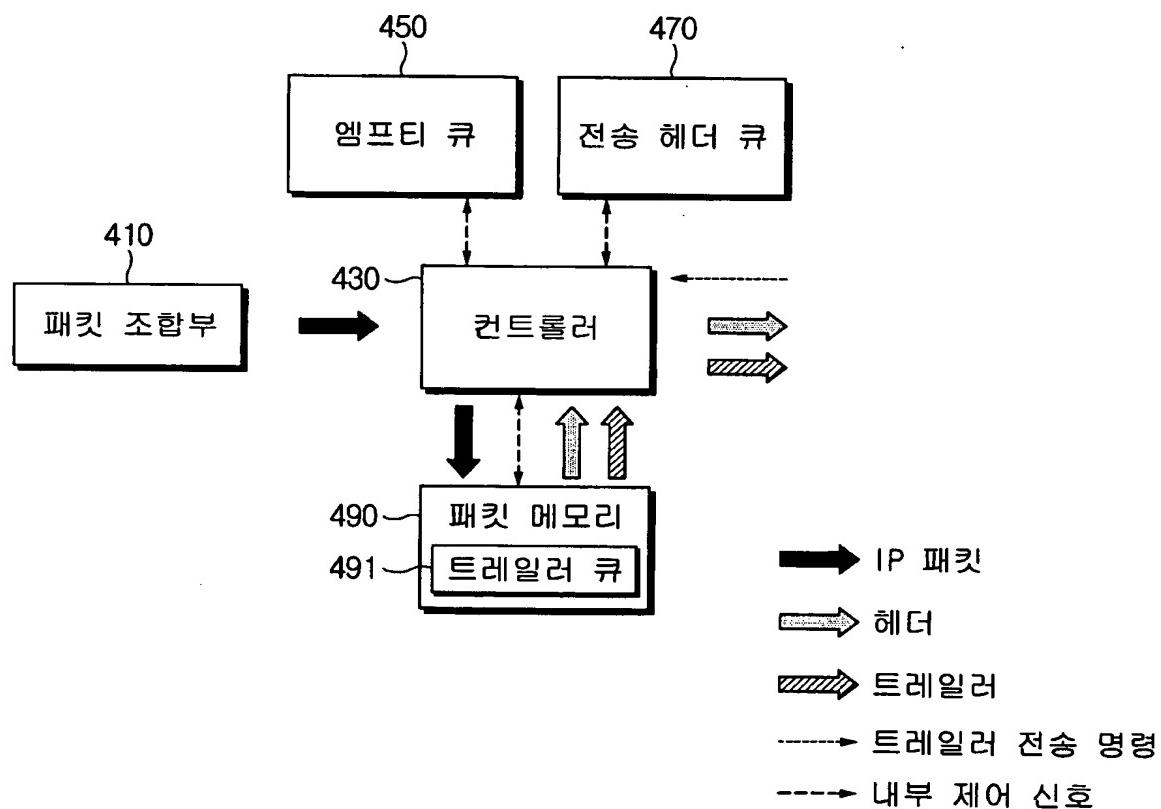
【도 2】



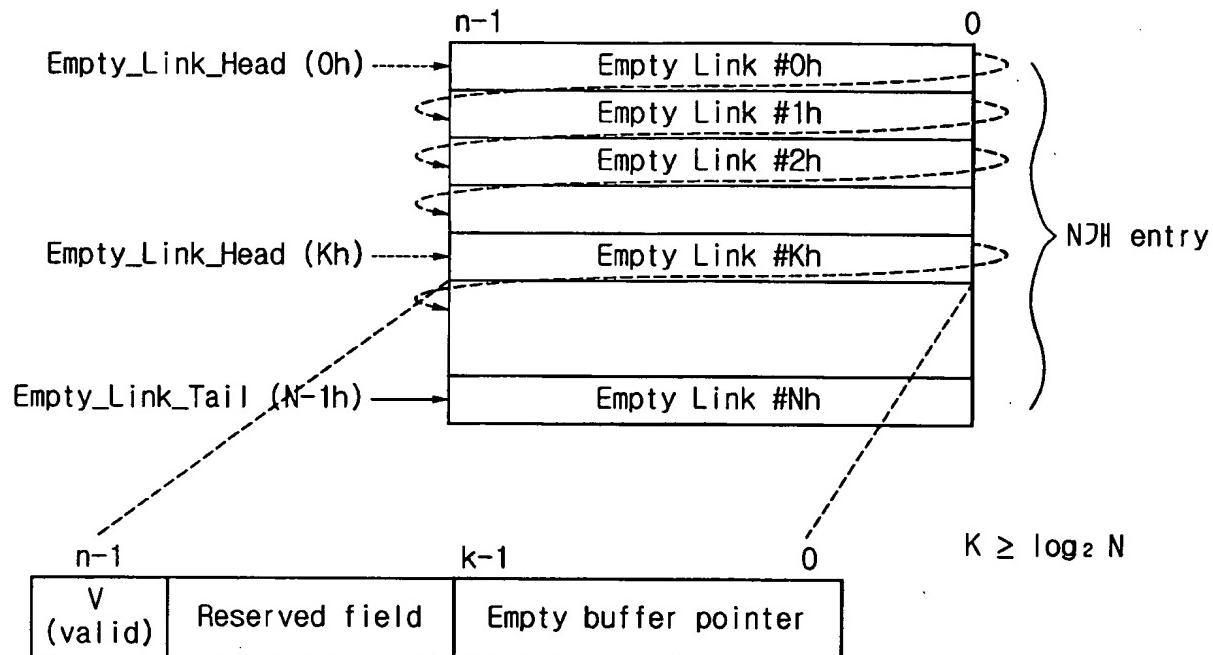
【도 3】



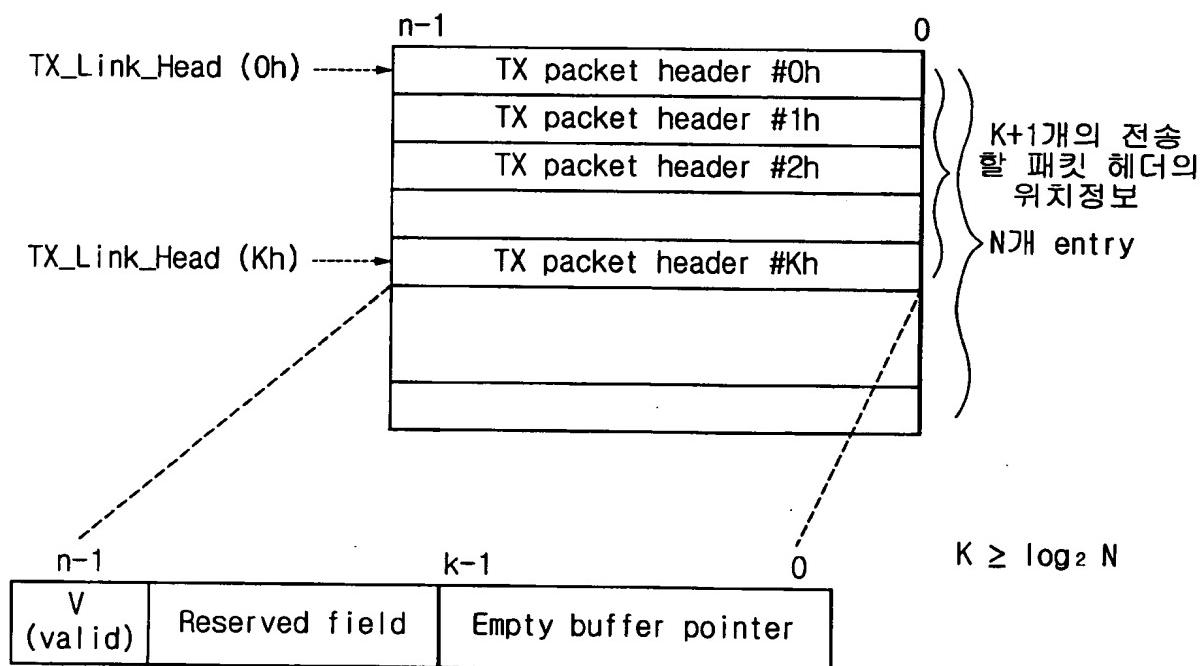
【도 4】



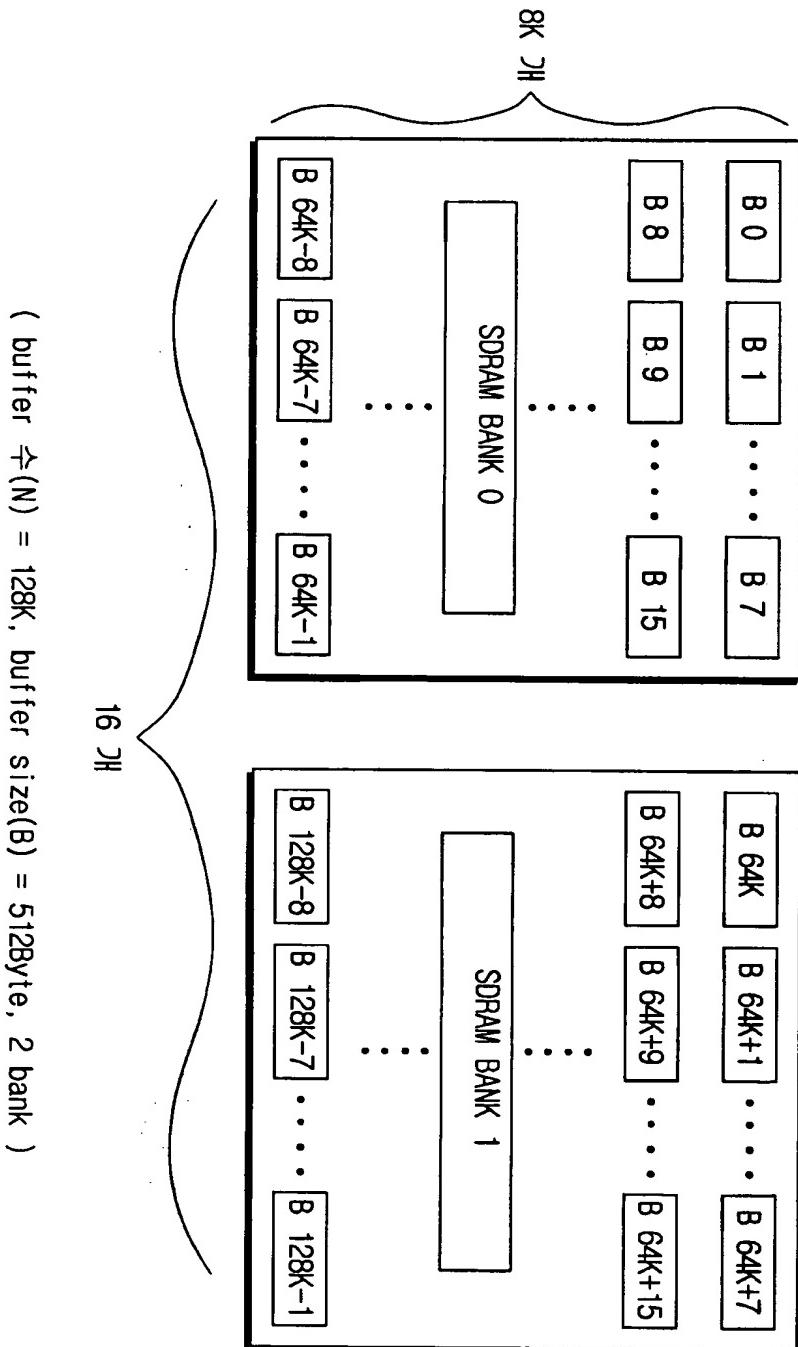
【도 5】



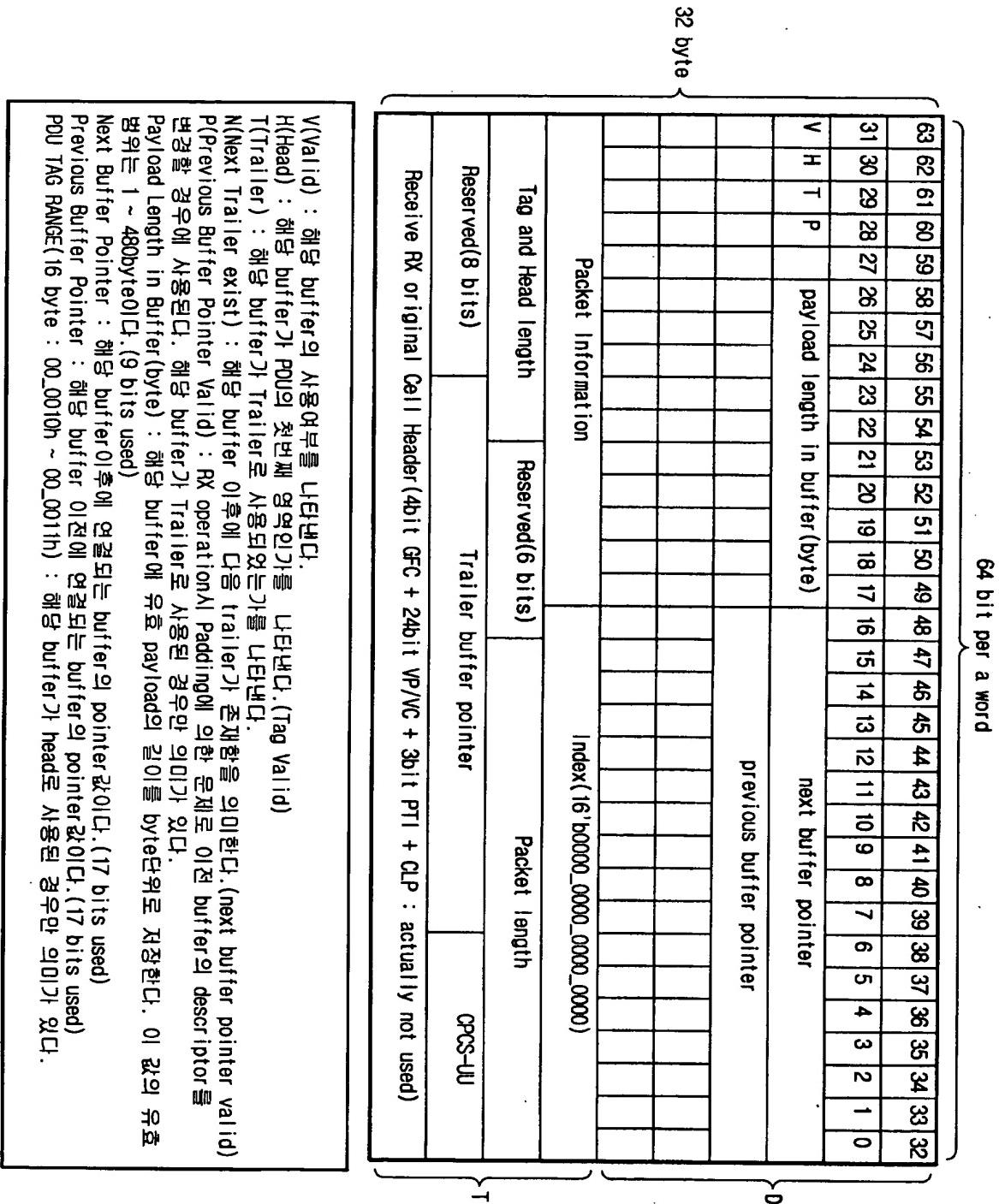
【도 6】



【도 7a】

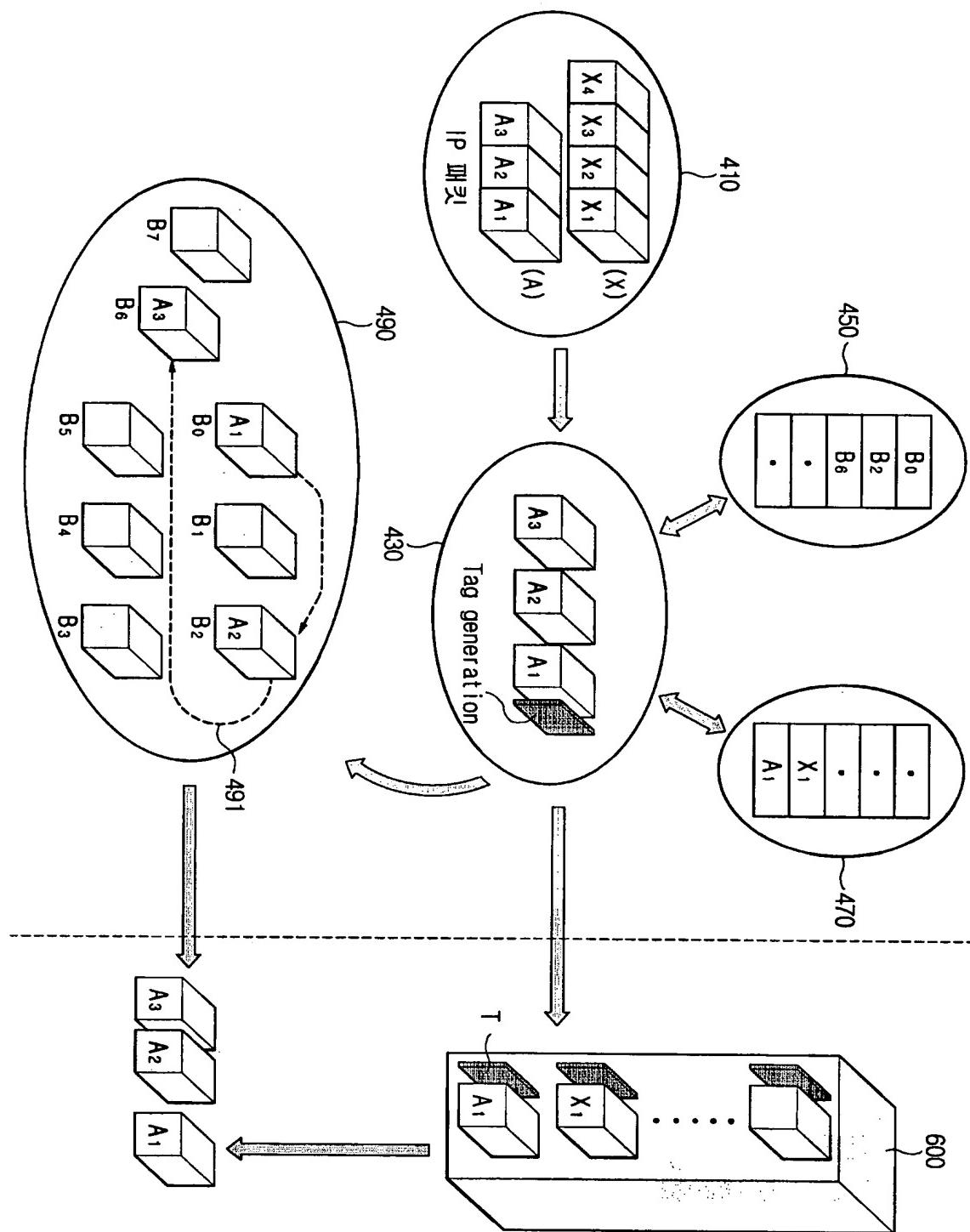


### 【도 7b】



**H(Valid)** : 해당 buffer의 사용여부를 나타낸다.  
**T(Trailer)** : 해당 buffer가 Trailer로 사용되었는가를 나타낸다.  
**N(Next Trailer exist)** : 해당 buffer 이후에 다음 trailer가 존재함을 의미한다.(next buffer pointer valid)  
**P(Previous Buffer Pointer Valid)** : RX operation시 Padding에 의한 문제로 이전 buffer의 descriptor를 변경할 경우에 사용된다. 해당 buffer가 Trailer로 사용된 경우만 의미가 있다.  
**Paylod Length in Buffer(byte)** : 해당 buffer에 유효 payload의 길이를 byte단위로 설정한다. 이 값의 범위는 1 ~ 480byte이다.(9 bits used)  
**Next Buffer Pointer** : 해당 buffer이후에 연결되는 buffer의 pointer값이다.(17 bits used)  
**Previous Buffer Pointer** : 해당 buffer 이전에 연결되는 buffer의 pointer값이다.(17 bits used)  
**POU TAG RANGE(16 byte : 00\_0010h ~ 00\_0011h)** : 해당 buffer가 header 사용된 경우만 의미가 있다.

【도 8】



## 【도 9】

